Programming Assignment1: Binary Decision Trees

Due: 11:00am Oct. 6th

In this programming assignment, your task is to implement one of the common machine learning algorithms: Decision Trees. You will train and test a binary decision tree with the dataset we provided.

Part 1. Implement a binary decision tree learning algorithm. [30 points]

There are different ways to design a decision tree. In this assignment, you should simply pick one feature to split on, and determine the threshold value to use in the split criterion for each non-leaf node in the tree. The optimal split at each node should be found using the information gain criterion taught in class(see the slides for Lecture 2). Also, since you are asked to build a binary decision tree, you should do only binary splits. That means, each split should simply determine if the value of a particular feature in the feature vector of a sample is less than or equal to a threshold value or greater than the threshold value. Please note that the features in the provided dataset are continuously valued. So you need to think about how to chose and compare possible threshold values when there is an infinite number of possible feature values.

You should name your main script BinaryDecisionTree.py which accepts three arguments and outputs your predictions in PredictY.csv. The grader will run your code on the command line in the following manner:

>python BinaryDecisionTree.py TrainX.csv TrainY.csv TestX.csv

Your code should then learn a binary decision tree using the training set TrainX.csv and TrainY.csv, and then make predictions for all the samples in TestX.csv and output the labels to PredictY.csv.

Part 2. Evaluate and draw the binary decision tree [10 points]

Draw by hand the decision tree you just trained on the training set. If the tree is too big or too complicated, you can stop at depth 4 of the tree(The root node of the tree has a depth of 0.). What is the total number of nodes in the tree? What is the total number of leaf nodes in the tree? What are the classification accuracies on the training set and the test set, respectively?
Note: You do note need to write code to draw a diagram of the tree. But you do need to write code for other questions.

Part 3. Experiments [10 points]

Train your binary decision tree with increasing sizes of training set, say 10%, 20%, …, 100%. and test the trees with the test set. Make a plot to show how training and test accuracies vary with number of training samples.

Dataset:

The dataset we use is the Wisconsin Diagnostic Breast Cancer (WDBC) dataset. We have split the date set into training set and test set stored in four csv files. The dataset can be downloaded on the course website. (http://www.cs.princeton.edu/courses/archive/fall16/cos402/ex/trainX.csv, http://www.cs.princeton.edu/courses/archive/fall16/cos402/ex/trainY.csv, http://www.cs.princeton.edu/courses/archive/fall16/cos402/ex/testX.csv, http://www.cs.princeton.edu/courses/archive/fall16/cos402/ex/testY.csv)
TrainX.csv has 455 samples, TrainY.csv has labels for the samples in TrainX.csv. Similarly, TestX.csv has 57 samples, TrainY.csv has labels for the samples in TrainX.csv. Each row in TrainX.csv or TestX.csv representing a sample of biopsied tissue. The tissue for each sample is imaged and 10 characteristics of the nuclei of cells present in each image are characterized. These characteristics are
1. Radius
2. Texture
3. Perimeter
4. Area
5. Smoothness

6. Compactness
7. Concavity
8. Number of concave portions of contour
9. Symmetry
10. Fractal dimension

Each sample used in the dataset (TrainX.csv and TestX.csv) is a feature vector of length 30. The first 10 entries in this feature vector are the mean of the characteristics listed above for each image. The second 10 are the standard deviation and last 10 are the largest value of each of these characteristics present in each image.
Each sample is also associated with a label provided in TrainY.csv or TestY.csv. A label of value 1 indicates the sample was for malignant (cancerous) tissue. A label of value 0 indicates the sample was for benign tissue.

What and how to turn in:

1. Turn in hard copies in class on the due date.

A printout of all your python scripts and answers to questions in all sections.

2. Upload your code to CS dropbox by the due date.

Using this DropBox link,http://dropbox.cs.princeton.edu/COS402_F2016/Programming_Assignment1, upload all your python scripts. You should only turn in uncompressed .py files. All code should be working and well documented. If appropriate, a readme.txt file explaining briefly how your code is organized, what data structures you are using, or anything else that will help the graders understand how your code works.